Browserul lui Biju

Responsabili tema:
Alexandru-Gabriel Bejan
Adrian-Emanuel Dicu
Theodor Oprea
Andrei Simescu
Ioan-Teodor Teugea

Deadline:
16.01.2022, ora 23:55
”Timpul trece, voi?”

Cuprins

0 Backstory-ul lui Biju

Biju ar face orice pentru regina lui. Recent, ea a citit un articol pe Internet despre problemele motorului de căutare Google si a browserului Chrome legate de intimitate si colectarea de date a utilizatorilor. Astfel, ea l-a rugat pe Biju sa rezolve problema asta. Soluția lui a fost să vă angajeze pentru a crea un browser si un motor de căutare, promițându-vă mulți bani (să vadă tot Israelul că îl ține portofelul) și că va veni să cânte la două evenimente la care îl chemați.

0.1 Datele de intrare

Site-urile, datele și informațiile lor sunt stocate în fișiere (câte un fișier pentru fiecare site). Un fișier va fi alcătuit din caractere ASCII și va conține URL-ul site-ului, lungimea în octeți a codului HTML, numărul de accesări, checksum-ul (în această ordine), apoi codul HTML al site-ului. Codul începe cu un tag <html> și se termină cu un tag </html>. În interiorul codului poate apărea următoarele taguri, unul singur din fiecare (fiecare tag are o variantă de început <tag> și o variantă de sfârșit </tag>):

- <title> - delimitează titlul paginii (va fi folosit în tab-ul din browser, doar unul per site)
- <p> - indică un paragraf în site

Tagul <p> poate indica folosind CSS culoarea textului
("color: [culoare text]") și culoarea fundalului: pentru elementul respectiv ("background-color: [culoare fundal]"), separate prin punct și virgulă. Culorile pot fi: white, black, red, green, blue, yellow. Dacă nu există CSS pentru un element, culoarea textului va fi neagră, iar cea de fundal albă.

Un tag <p> se poate întinde pe mai multe linii!

De asemenea, va exista un fișier principal numit "master.txt" care va contine numele fiecărui fișier corespunzător unui site.

Exemplu de date de intrare:

master.txt

```
site1.html
site2.html
site3.html
```

site1.html

```
www.upb.ro 234 0 0
<html>
<title>Universitatea Politehnica Bucuresti</title>
<p style="color:red;background-color:blue;">Bine ati venit rau ati nimerit
Facultati:
- Automatica si Calculatoare
- ETTI
- Altele, de care nu intereseaza pe nimeni</p>
</html>
```

site2.html

```
https://bit.ly/3dCfNBj 63 1108062718 0
<html>
<title>Important</title>
<p>FOARTE IMPORTANT</p>
</html>
```

site3.html

```
www.a-s-e.ro 134 8008135 0
<html>
<title>ASE</title>
<p>Turul facultatii:
https://bit.ly/3GtFlwN
Comunicat de presa important:
https://bit.ly/3rPuEkm</p>
</html>
```

Important: Pentru reprezentarea in memorie unui site, se va folosi o structura care sa contina URL-ul, numarul de accesari, checksum-ul, titlul si continutul, obtinute dupa parsarea codului HTML. URL-ul si titlul vor avea o dimensiune maxima de 50 de caractere, iar continutul va avea exact dimensiunea necesara pentru stocarea sa.

Pentru culoarea textului si culoarea de fundal, se vor folosi enumerari.

O linie dintr-un fisier va avea maxim 100 de caractere, iar numele unui fisier va avea maxim 30 de caractere. Datele corespunzatoare tuturor site-urilor se vor memora intr-un vector de astfel de structuri, alocat dinamic cu o marime initiala de 3 elemente si realocat daca este nevoie, marit de fiecare data cu inca 3 elemente.

1 Task 1

Pentru a putea să facem ceva cu site-urile, mai întâi trebuie să încărcăm în memorie baza de date. Pentru a îi arăta lui Biju că am încărcat datele bine, vom afișa URL-ul site-urilor, numărul vizualizărilor și titlul, în ordinea in care apar in master.txt. (0.4p)

Exemplu:

Input:

Exemplul prezentat anterior

Output:

```
www.upb.ro 0 Universitatea Politehnica Bucuresti
https://bit.ly/3dCfNBj 1108062718 Important
www.a-s-e.ro 8008135 ASE
```

2 Task 2

Pentru că Biju este un om ocupat, nu a vrea să stea sa caute prin baza de date, când îşi cauta inspiraţie pe la turci de manele noi, aşa că are nevoie de ajutorul vostru pentru a-l ajuta cu asta. Va trebui să citiți de la tastatură cuvinte cheie separate printr-un spațiu până la întâlnirea caracterului '\n' și să afișați site-urile care au în conținutul lor cel puțin unul dintre cuvintele cheie. Site-urile găsite se vor ordona alfabetic, iar în cazul în care vreun site a facut manevra lui Biju şi l-a copiat pe altul (2 site-uri au titlul identic), atunci al doilea criteriu de ordonare, va fi numarul de accesari în ordine descrescătoare. (0.3p)

Important: La taskurile 2 si 3, ordonarea se va face folosind o funcție de sortare (aceeasi la ambele) care primește ca argument un pointer la o funcție comparator (cate una diferita pentru fiecare task).

2.1 Exemplul 1

Baza de date (dupa incarcarea in memorie):

```
https://bit.ly/3Gv2Iq1 8650074 Vremea Bucuresti 8 Decembrie
alabalaportocala.eu 12 Acesta este un site foarte frumos
upb.ro 0 Bine ati venit rau ati nimerit
facebook.com 12302 Bine ati venit pe Facebook!
a-copy-book.com 10 Bine ati venit pe Facebook!
```

Input:

```
Bine ati venit
```

Output:

```
facebook.com
a-copy-book.com
upb.ro
```

S-au selectat în acest caz facebook.com, a-copy-book.com şi upb.ro, deoarece alabalaportocala.eu şi https://bit.ly/3Gv2Iq1 8650074, nu conţin niciunul din termenii: "Bine","ati" sau "venit". În cazul site-urilor facebook.com şi a-copy-book.com, observăm că au aceeaşi descriere, aşa că le-am sortat descrescător în funcţie de numărul de accesări.

2.2 Exemplul 2

Baza de date:

```
https://bit.ly/3dzxOQE 43259710 Merg cu 30 la ora pe langa un urs brun
gigica.eu 12 Acesta este un site frumos rau
upb.ro 0 Bine ai venit rau ai nimerit
facebook.com 12302 Bine ati venit pe Facebook!
```

Input:

```
rau pe
```

Output:

```
gigica.eu
facebook.com
upb.ro
https://bit.ly/3dzxOQE
```

Se afişează toate site-urile, deoarece gigica.eu şi upb.ro conţin cuvântul “rău”, iar https://bit.ly/3dzxOQE şi facebook.com conţin “pe”.

2.3 Exemplul 3

Baza de date:

```
https://bit.ly/07n4mc4rt3l4 804 Cine nu are site să isi faca
alabalaportocala.eu 12 Acesta este un site foarte frumos
upb.ro 0 Bine ati venit rau ati nimerit
facebook.com 12302 Bine ati venit pe Facebook
```

Input:

```
candva avion bani
```

Output:

Nu se va afisă nimic, deoarece niciun site nu conţine cuvintele căutate.

3 Task 3

După ce s-a trezit din mahmureala de după parțialul la ALGAED, cum încă îi este greu să navigheze printre rezultatele căutării simple, Biju a decis să filtreze căutările după un set de reguli inspirat de cei de la Google, precum excluderea unor cuvinte sau căutarea unei secvențe într-o ordine exacta. (0.3p)

Input: Fișierele specifice pentru site-uri și un query de căutare

Output: Site-urile ordonate descrescător după numărul de accesări.

Important: La taskurile 2 si 3, ordonarea se va face folosind o funcție de sortare (aceeasi la ambele) care primește ca argument un pointer la o funcție comparator (cate una diferita pentru fiecare task).

3.1 Exemplul 1

Excluderea cuvintelor se realizează prin plasarea caracterului ‘-’ în fața acestora. Se poate exclude un singur cuvant per cautare. De exemplu, căutarea:

```
bine ați venit -rau
```

va afișa toate site-urile care conțin unul din cuvintele “bine”, “ați”, “venit” dar care nu conțin cuvântul “rău”.

master.txt

```
site1.html
site2.html
site3.html
site4.html
```

site1.html

```
https://bit.ly/3Gv2Iq1 82 8650074 0
<html>
<title> Ceva frumos </title>
<p> Vremea Bucuresti 8 Decembrie </p>
</html>
```

site2.html

```
alabalaportocala.eu 89 12 0
<html>
<title>Alabalaportocala</title>
<p> Acesta este un site foarte frumos </p>
</html>
```

site3.html

```
upb.ro 73 0 0
<html>
<title>UPB</title>
<p> Bine ati venit rau ati nimerit </p>
</html>
```

site4.html

```
facebook.com 74 12302 0
<html>
<title>Facebook</title>
<p> Bine ati venit pe Facebook!</p>
</html>
```

Input:

```
Bine ati venit -rau
```

Output:

```
facebook.com
```

S-a selectat în acest caz doar facebook.com și nu upb.ro pentru că cel din urma conține și cuvântul "rau".

3.2 Exemplul 2

Căutarea unei secvențe într-o ordine exactă se realizează prin includerea secvenței respective între ghilimele, "secvența".

De exemplu, căutarea:

```
frumos "ati venit"
```

va afișa toate site-urile în care se găsește cuvântul frumos, dar și cele care conțin secvența "ati venit".

master.txt

```
site1.html
site2.html
site3.html
site4.html
```

site1.html

```
https://bit.ly/3dzxOQE 83 43259710 0
<html>
<title> SSh </title>
<p> Merg cu 30 la ora pe langa un urs brun </p>
</html>
```

site2.html

```
alabalaportocala.eu 89 12 0
<html>
<title>Alabalaportocala</title>
<p> Acesta este un site foarte frumos </p>
</html>
```

site3.html

```
upb.ro 71 0 0
<html>
<title>UPB</title>
<p> Bine ai venit rau ai nimerit </p>
</html>
```

site4.html

```
facebook.com 74 12302 0
<html>
<title>Facebook</title>
<p> Bine ati venit pe Facebook!</p>
</html>
```

Input:

```
frumos "ati venit"
```

Output:

```
facebook.com
alabalaportocala.eu
```

Se afișează doar alabalaportocala.eu pentru că în cadrul acestuia apare cuvântul frumos si facebook.com deoarece conține șirul “ați venit”, însă upb.ro nu se afișează pentru că deși conține cuvântul “venit”, acesta nu conține secvența “ați venit”

3.3 Exemplul 3

Aceste 2 reguli se pot folosi și împreuna, afișând-se site-urile care nu conțin cuvintele de după ‘-’ și care conțin secvențele din ghilimele în ordine exactă.

De exemplu, căutarea:

```
frumos "ati nimerit" -rau
```

va afișa toate site-urile care conțin fie cuvântul “frumos”, fie secvența “ați nimeri”, și care nu conține cuvântul “rau”.

master.txt

```
site1.html
site2.html
site3.html
site4.html
```

site1.html

```
https://bit.ly/30crKL6 81 1107800804 0
<html>
<title>Three letters back</title>
<p> Nu ati nimerit rău aici </p>
</html>
```

site2.html

```
alabalaportocala.eu 89 12 0
<html>
<title>Alabalaportocala</title>
<p> Acesta este un site foarte frumos </p>
</html>
```

site3.html

```
upb.ro 73 0 0
<html>
<title>UPB</title>
<p> Bine ati venit rau ati nimerit </p>
</html>
```

site4.html

```
facebook.com 74 12302 0
<html>
<title>Facebook</title>
<p> Bine ati venit pe Facebook!</p>
</html>
```

Input:

```
frumos "ati nimerit" -rau
```

Output:

```
alabalaportocala.eu
upb.ro
```

Se afișează doar alabalaportocala.eu pentru că se găsește cuvântul “frumos” în cadrul acestuia (și nu conține “rău”) și upb.ro deoarece conține secvența “ați nimeri” (și nu conține “rău”). Însă upb.ro se omite pentru că, deși conține secvența exactă “ați venit”, acesta conține și cuvântul “rău”, care trebuie omis.

4 Task 4

Dupa ce a pierdut suficient timp pe retelele de socializare, Biju isi reaminteste ca din senin de cursul de structuri de date din facultate in care a invatat despre functii hash criptografice. Speriat de ideea ca acum nu doar marile corporatii ii pot fura datele personale, dar si siteuri malitioase, el se hotaraste sa isi puna in aplicare skillurile invatate pentru a lua restanta la SD ca sa determine daca site-urile pe care el le-a accesat pana acum chiar sunt legite sau nu, aplicand o functie hash pe continutul lor si verificand-o cu checksum-ul oficial.

Pentru inceput, reprezentam continutul unui site ca pe un sir de caractere S. Fiecare caracter din S are o reprezentare binara pe 1 octet (8 biti) in functie de codul ASCII corespunzator (a se vedea Tabelul ASCII). (0.3p)

Definim urmatoarele 2 functii:

```
int rotr(char x, int k); // roteste la dreapta bitii lui x cu k pozitii
int rotl(char x, int k); // roteste la stanga bitii lui x cu k pozitii
```

De exemplu, daca un caracter x are reprezentarea binara: ”01100101” atunci $rotr(x, 2)$ va fi ”01011001”, respectiv rotl(x, 4) va fi ”01010110”.

Checksum-ul website-ului este egal cu:

```
rotl(S[0], 0) ^ rotr(S[1], 1) ^ rotl(S[2], 2) ^ .. ^ rotl/rotr(S[n-1], n-1)
```

Cu alte cuvinte, fiecarui caracter de pe o pozitie arbitrara p este rotit cu p pozitii la stanga sau la dreapta in functie de paritatea lui p (la stanga daca p este par, altfel la dreapta). La final, se aplica operatorul XOR (operatorul $\wedge$) tuturor valorilor si se obtine un numar care este verificat cu checksum-ul oficial.

De la standard input se vor citi cat timp se poate numele site-urilor care trebuiesc verificate, cate un site pe fiecare linie. Fiecare site reprezinta un query si pentru fiecare query se va afisa cate o linie in standard output.

Daca site-ul nu exista in baza de date se va afisa mesajul: ”Website not found!”. Altfel, se verifica checksum-ul site-ului dat in standard input. Daca valoarea obtinuta este identica cu cea oficiala, se va afisa mesajul: ”Website safe!”, altfel se va afisa mesajul ”Malicious website! Official key: CHEIE_OFICIALA. Found key: CHEIE_GASITA.” unde CHEIE_OFICIALA si CHEIE_GASITA se vor inlocui cu valorile corespunzatoare.

Exemplu:

master.txt

```
site1.html
site2.html
site3.html
site4.html
```

site1.html

```
https://bit.ly/30crKL6 81 1107800804 126
<html>
<title>Three letters back</title>
<p> Nu ati nimerit rau aici </p>
</html>
```

site2.html

```
alabalaportocala.eu 89 12 217
<html>
<title>Alabalaportocala</title>
<p> Acesta este un site foarte frumos </p>
</html>
```

site3.html

```
upb.ro 73 0 206
<html>
<title>UPB</title>
<p> Bine ati venit rau ati nimerit </p>
</html>
```

site4.html

```
facebook.com 74 12302 84
<html>
<title>Facebook</title>
<p> Bine ati venit pe Facebook!</p>
</html>
```

Input:

```
https://bit.ly/30crKL6
alabalaportocala.eu
upb.ro
facebook.com
notfound.com
```

Output:

```
Website safe!
Malicious website! Official key: 217. Found key: 199
Website safe!
Malicious website! Official key: 84. Found key: 74
Website not found!
```

Explicatie:
Pentru primul website continutul lui este string-ul:

```
<html>\n<title>Three letters back</title>\n<p> Nu ati nimerit rau aici </p>\n</html>
```

Atentie! mai sus, \n este un singur caracter (newline).
Astfel, checksum-ul va fi egal cu:

```
60 ^ 52 ^ 209 ^ 173 ^ 198  ... ^ 91 ^ 216 ^ 62 = 126
```

5 Task 5

In final, motorul are nevoie si de o interfață grafică pentru a fi ușor de utilizat. Pentru aceasta veți folosi biblioteca ncurses. Aceasta permite crearea unor interfețe grafice simple in terminal. Interacțiunea cu această interfață se va face doar prin intermediul tastelor, de aceea in partea jos a fiecărei pagini va trebui sa includeți o legendă cu comenzile posibile si tastele aferente. (0.7p)

5.1 Pagina de căutare

Pentru început, va trebui să construiți un ecran ce va conține numele motorului de căutare. Odată cu apăsarea tastei de căutare, "C", va apărea o bară de căutare iar cursorul aplicației va fi mutat la începutul acesteia. Utilizatorul poate acum scrie șirul de căutare, iar atunci cand a terminat va apăsa "Enter", tastă ce blochează câmpul de căutare. Mai departe, pentru a efectua o căutare simplă cu textul introdus se va apăsa tasta "S", iar pentru una avansată tasta "A". Tasta "Q" va închide programul.

5.2 Pagina de rezultate

După lansarea unei căutari se va afișa o nouă fereastră cu rezultatele acesteia. Aceasta va conține in partea de sus o bară cu textul căutat, apoi o listă formata din URL-ul si titlul paginilor, asemănător Google. Pentru acest task se recomandă folosirea bibliotecii de meniuri din ncurses, pentru a face mai ușoară selectarea paginilor. Tasta "B" poate fi folosită pentru a ne întoarce la pagina de căutare, iar pagina dorită poate fi selectată cu tasta "Enter".

5.3 Pagina web

Dupa selectarea unui rezultat va trebui sa afișați pagina, conform HTML-ului paginii. Astfel, titlul va fi in partea de sus a paginii si va fi bold, urmat de fiecare paragraf ce va fi afișat individual conform specificațiilor de stil. Tasta "B" poate fi folosită pentru a ne întoarce la pagina cu rezultatele.

5.4 Precizari ncurses

- Pentru obtinerea bibliotecii ncurses (pe o distributie de Linux bazata pe Debian, instalati cu: apt-get install libncurses5-dev).
- Detalii despre biblioteca ncurses se gasesc in urmatoarele materiale:
  - Documentatia oficiala
  - Scurta introducere in utilizarea bibliotecii
  - Exemple

6 Restrictii si precizari generale

- Se vor aplica urmatoarele depunctari:
  - 15% din punctajul obtinut daca nu eliberati memoria alocata dinamic (puteti folosi utilitarul valgrind pentru a verifica acest lucru). Este posibil ca biblioteca ncurses sa aiba leak-uri de memorie. Nu veti fi depunctati pentru acestea.
  - 10% din punctajul obtinut daca nu inchideti fisierele dupa ce nu le mai utilizati.
  - 10% din punctajul obtinut pentru un coding style necorespunzator
  - 10% din punctajul obtinut pentru un README necorespunzator
- Tema va fi rezolvata obligatoriu in limbajul C. Nu folositi elemente ale limbajului C++.
- Fiecare problema va fi rezolvata intr-un fisier separat, numit taskX.c cu X de la 1 la 4. Task-ul 5 va fi rezolvat intr-un fisier numit browser.c.
- Puteti (si este recomandat) sa aveti alte fisiere .c si .h in care sa implementati functionalitate comuna tuturor task-urilor.
- Veti trimite o arhiva ZIP cu numele de tipul GRUPA_Nume_Prenume_Tema2.zip. De exemplu: 311CC_Popescu_Maria_T care va contine fisierele necesare, Makefile si README. Fisierele trebuie sa fie in radacina arhivei, nu in alte subdirectoare.
- In README precizati cat timp v-a luat implementarea cerintelor si explicati, pe scurt, implementarea temei (comentariile din cod vor documenta mai amanuntit rezolvarea) Este recomandat ca liniile de cod si cele din fisierul README sa nu depaseasca 80 de caractere.
- Folositi un coding style astfel incat codul sa fie usor de citit si inteles. De exemplu:
  - Dati nume corespunzatoare variabilelor si functiilor.

  - Nu adaugati prea multe linii libere intre instructiuni sau la sfarsitul fisierului). Principalul scop al spatiilor este identarea.
  - Fiti consecventi in coding style-ul ales. Va recomandam sa parcurgeti aceasta resursa.
  - Exista programe sau extensii pentru editoare text care va pot formata codul. Desi va pot ajuta destul de mult, ar fi ideal sa incercati sa respectati coding style-ul pe masura ce scrieti codul.

- **Temele sunt strict individuale. Copierea temelor va fi sanctionata. Persoanele cu portiuni de cod identice nu vor primi niciun punctaj pe tema.**
- Temele trimise dupa deadline nu vor fi luate ın considerare. Nu se accepta teme trimise prin alte mijloace decat prin vmchecker.